ISSN: 2278-0181
Vol. 2 Issue 12, December - 2013

Analysis and Comparison on Image Restoration Algorithms Using MATLAB

Admore Gota
School of Electronics Engineering, Tianjin University of Technology and Education (TUTE), Tianjin P.R China.

Zhang Jian Min
School of Electronics Engineering, Tianjin University of Technology and Education (TUTE), Tianjin P.R China.

Abstract

TheImage restoration is the recovery of an image that has been degraded by blur and noise. Degradation typically involves blurring of the original image and corruption noise. The recovery of an original image from degraded observations is of paramount importance and can find its application in several scientific areas including medical and diagnostics, military surveillance,satellite and astronomical imaging, remote sensing, authentication automated industry inspection and many areas. Image restoration assures good insights of image when it is subjected to further techniques of image processing.

Presentation of the results and comparisons of the restoration algorithms namely Weiner Filter, the Regularized filter and Lucy-Richardson as implemented in Matlab. This is done using an image that is degraded by motion blur with Gaussian, Salt and Pepper and Speckle noise models respectively and restoration of same image using various algorithms mentioned above. Image restoration algorithms, distinguish themselves from enhancement methods in that they are based on models for degrading process and for the ideal image. The image restoration methods in this paper fall under the class of linear spatially invariant restoration filters. Degradation typically involves blurring of the original image and corruption noise.

1. Introduction

Visual information transmitted in the form of digital images is becoming a major method of communication in the modern age. Images are produced to record or display useful information. However, due to imperfections in the imaging and capturing process, the recorded image invariably represents a degraded version of the original scene. The undoingof these imperfections is crucial to many of the subsequent image processing tasks. A wide range of degradations such as noise, geometrical degradations, illumination and color imperfections and blur. This paper concentrates on basic methods for removing blur and noise, in this case Gaussian, Salt and Pepper and Speckle noise from recorded and spatially images.

The field of image restoration which is sometimes referred to as image deblurring or image deconvolution is concerned with the reconstruction or estimation of uncorrupted image from blurred and noisy one. Image restoration is associated with minimizing or even removing artifacts due to blurring and noise. Blurring which is a linear form of degradation can occur due to camera defocussing or due to motion. This project concentrates on the Analysis and Comparison of methods of image restoration algorithms.

2. The Objectives of the project are:

- To find out a suitable highly accurate restoration algorithm to filter and remove the degradation on an image using Matlab simulation.
- To investigate the strength and limitations of each image restoration algorithm.
- To find a suitable algorithm with high performance of filtering and doing a great job of restoring the image.
- To generate an estimate of the original image prior to the degradation.

3. Blur Models

In most cases the blurring of images is a spatially continuous process. Since identification and restoration algorithms are always based on spatially discrete images, we present the blur models in their continuous forms. The imperfections in the image formation process are modelled as passive operations on the data that is no energy is absorbed or generated. Following is the presentation of four common point-spread functions, which are encountered regularly in practical situations of interest.

A.No Blur

In case the recorded image is imaged perfectly, no blur will be apparent in the discrete image.

B.Linear motion blur

Many types of motion blur can be distinguished all of which are due to relative motion between the recording device and the scene. This can be in the form of a translation, a rotation, a sudden change of scale. Here only the important case of a global translation will be considered.

C.Uniform Out of focus blur

When a camera images a 3-D scene onto a 2-D imaging plane, some parts of the scene are in focus while other parts are not. If the aperture of the camera is circular, the image of any point source is a small disk, known as the circle of confusion (COC).

D.Atmospheric turbulence blur

Atmospheric turbulence is a severe limitation in remote sensing. Although the blur introduced by atmospheric turbulence depends on a variety of factors such as temperature, wind speed and exposure time, for long-term exposures the point spread function can be described reasonably well by a Gaussian function.

4. Noise Models

The noise commonly present in an image. Note that noise is undesired information that contaminates the image. In the image de-noising process, information about the type of noise present in the original image plays a significant role. Typical images are corrupted with noise modelled with either a Gaussian, or salt and pepper distribution. Another typical noise is a speckle noise, which is multiplicative in nature. Noise is present in an image either in an additive or multiplicative form.

A.Gaussian Noise

Its evenly distributed over the signal. This means that each pixel in the noisy image is the sum of the true pixel value and a random Gaussian distributed noise value. As the name indicates, this type of noise has Gaussian distribution, which has a bell-shaped probability distribution function given by,

$$F(g) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(g-m)^2/2\sigma^2}$$

Where g represents the gray level is the mean or average of the function, and σ *is* the standard deviation of the noise. Graphically it is represented as shown in Figure 1.When introduced into an image, Gaussian noise with zero mean and variance as 0.05 would look as in figure 2a. Figure 2b illustrates the Gaussian noise with mean (variance) as 1.5 (10) over a base image.

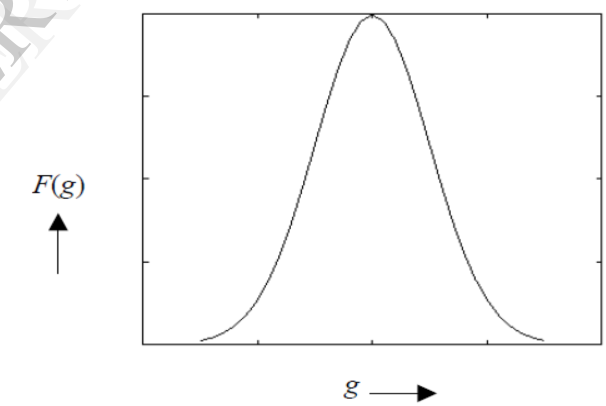
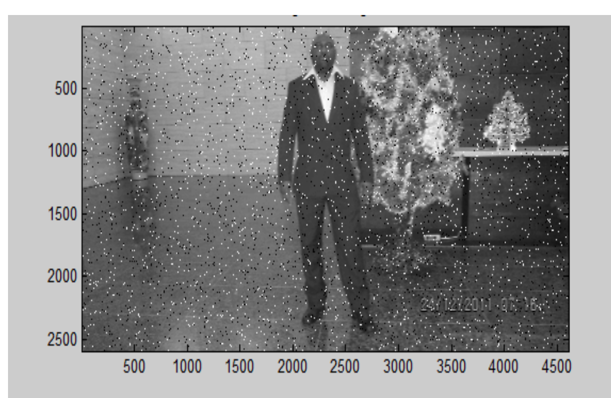
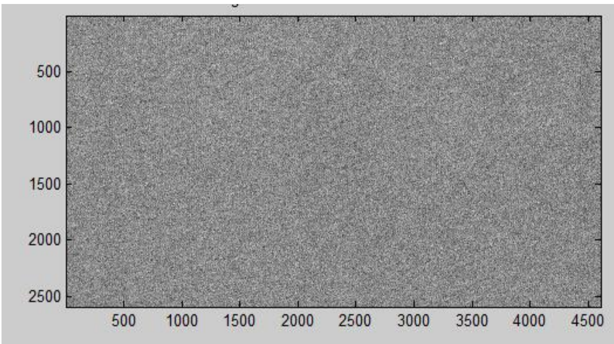


Figure 1. Gaussian Distribution



(a) (mean=0, variance =0.05)



(b) (mean=1.5, variance=10)

Figure 2: Image degraded by Gaussian noise with different mean and variance values

B. Salt and Pepper Noise

Is an impulse type of noise, which is also referred to as intensity spikes. This is caused generally due to errors in data transmission. It has only two possible values, a and b. The probability of each is typically less than 0.1. The corrupted pixels are set alternatively to the minimum or to the maximum value, giving the image a salt and pepper like appearance. Unaffected pixels remain unchanged. For an 8-bit image, the typical value for pepper noise is 0 and for salt noise 255.The salt and pepper noise is generally caused by malfunctioning of pixel elements in the camera sensors, faulty memory locations, or timing errors in the digitization process. The probability density function for this type of noise is shown in Figure 3. Salt and pepper noise with a variance of 0.05 is shown in Figure 4.

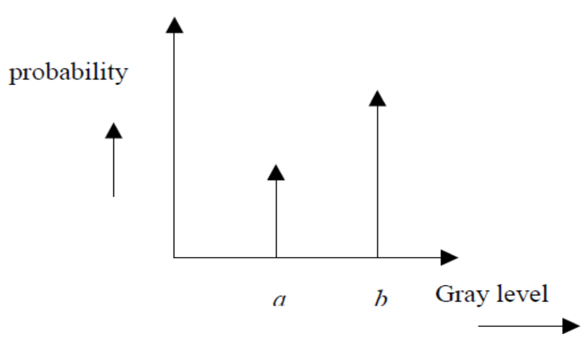
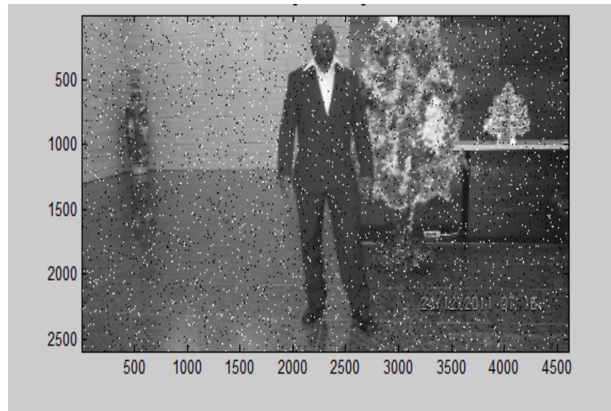


Figure 3: PDF for Salt and Pepper noise.



C. Speckle Noise

Is a multiplicative noise. This type of noise occurs in almost all coherent imaging systems such aslaser, acousticsand SAR (Synthetic Aperture Radar) imaginery. The source of this noise is attributed to random interference between the coherent returns. Fully developed speckle noise has the characteristic of multiplicative noise. Speckle noise follows a gamma distribution and is given as

$$F(g) = \frac{g^{\alpha-1}}{(\alpha-1)!\,a^{\alpha}} e^{-\frac{g}{a}}$$

The where variance is $a^2\alpha$ and$g$is the gray level.
The gamma distribution is given below in Figure 5. While the speckle noise (with variance 0.05) on an image looks as shown in Figure 6.

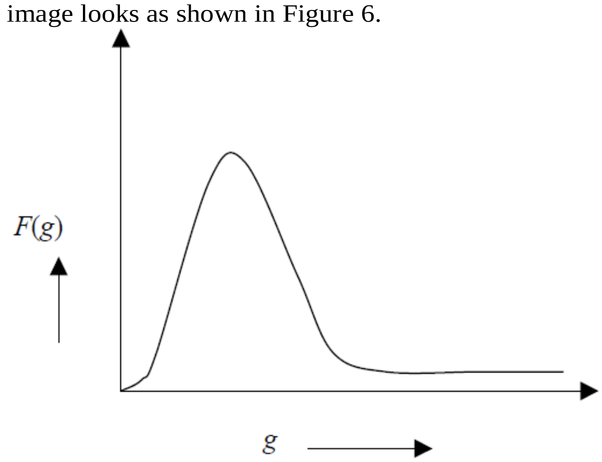


Figure 5: Gamma Distribution.

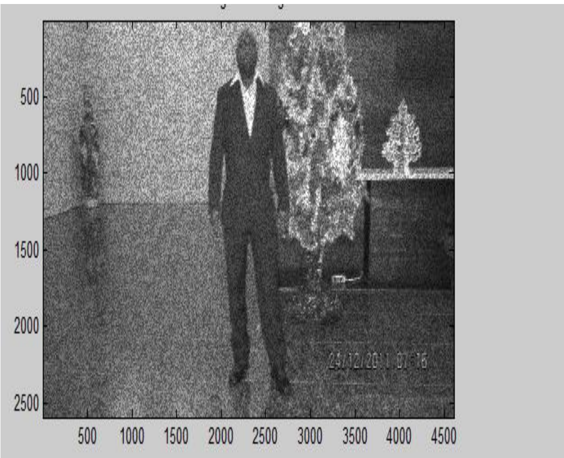


Figure 6: Image degraded by Speckle noise.

5. Restoration Algorithms

In image restoration the improvement in quality of the restored image over the recorded blurred one is measured by the signal-to-noise ratio improvement. When applying restoration filters to real images of which the ideal image is not available, often only the visual judgement of the restored image can be relied upon.

A. Wiener Filter

This filter can be used effectively when the frequency characteristics of the image and additive noise are known, to at least some degree. Wiener filters are often applied in the frequency domain. An important advantage of this algorithm is that it removes the additive noise and inverts the blurring simultaneously. A demerit of the Wiener filters is that they are unable to reconstruct frequency components which have been degraded by noise, but can only suppress them. These filters are comparatively slow to apply, since they require working in the frequency domain. The spatially truncated Wiener filter is inferior to the frequency domain version, but may be much faster.

B. Regularized Filter

Regularized filtering constraints are applied on the recovered image (e.g. smoothness) and limited information is known about the additive noise. The blurred and noisy image is restored by a constrained least square restoration algorithm that use a regularizedfilter. Although the Wiener filtering is the optimal tradeoff of inverse filtering and noise smoothing, in this case when the blurring filter is singular, the Wiener filtering actually amplify the noise.The implementation of the regularized inverse filter involves the estimation of the power spectrum of the original image in the spatial domain.

C. Lucy-Richardson algorithm

The Lucy-Richardson algorithm can be used effectively when the point-spread function PSF (blurring operator is known, but little or no information is available for the noise. The blurred and noisy image is restored by the iterative, accelerated, damped Lucy-Richardson algorithm. The additional optical system such as camera characteristics can be used as input parameters to improve the quality of the image restoration. The algorithm requires a good estimate of the process by which the image is degraded for accurate restoration. The degradation can be caused in many ways, such as subject movement, out-of-focus lenses, or atmospheric turbulence, and is described by the point spread function (PSF) of the system. The image is assumed to come from a Poisson process and therefore is corrupted by signal-dependent noise. There may also be electronic or quantization noise involved in obtaining the image.

6. Software Algorithm

```
% Digital Image Processing

        Clc;
        Close all;
        FltInitialCpuTime = cputime;

        ImgTemp = imread ('Gota.jpg', 'jpg');
        % Normalize Image
        imgTissue1 = double (imgTemp). / 255;

%----------------------------------------------------------
--------------------
% Part 1 - Convert image to gray level
        %[X, map] = rgb2ind (imgTissue1, 256);
        %imgTissue1 = ind2gray(X, map);
        %imwrite (imgTissue1, 'Gota.jpg', 'jpg');

%----------------------------------------------------------
--------------------
% Part 2 - Generation of Gaussian noise matrix
        vctTissue1Size = size(imgTissue1);
        MxGaussianNoise = 0.1.* randn
(vctTissue1Size (1), vctTissue1Size (2));
```

```
%--------------------------------------------------------------------------
% Part 3 - Point Spread Function (PSF) Generation
        filterPSF = fspecial('motion', 21, 11);
        imgTissue1Blur = imfilter(imgTissue1, filterPSF);
        imgTissue1BlurAndNoise = imadd (imgTissue1Blur, mxGaussianNoise);
        imwrite (imgTissue1BlurAndNoise, 'Gota_blur_and_noise.jpg', 'jpg');

%--------------------------------------------------------------------------
% Part 4 - Noise to Image Power Ratio
        imgTissue1Spectrum = abs(fft2(imgTissue1)).^2;
        fltTissue1Power = sum (imgTissue1Spectrum (:)) / numel (imgTissue1Spectrum);

        mxGaussianNoiseSpectrum = abs (fft2 (mxGaussianNoise)). ^2;
        fltGaussianNoisePower = sum (mxGaussianNoiseSpectrum (:)) / numel (mxGaussianNoiseSpectrum);

        fltNSR = fltGaussianNoisePower / fltTissue1Power;
        disp ('Noise to Signal Power Ratio:');
        disp (fltNSR);

%--------------------------------------------------------------------------
% Part 5 - Autocorrelation Functions
        mxTissue1Autocorrelation = fftshift(real(ifft2(imgTissue1Spectrum)));
        mxGaussianNoiseAutocorrelation = fftshift (real (ifft2(mxGaussianNoiseSpectrum)));

%--------------------------------------------------------------------------
% Part 6 - Wiener Filtering
        imgTissue1Weiner1 = deconvwnr (imgTissue1BlurAndNoise, filterPSF);
        imgTissue1Weiner2 = deconvwnr (imgTissue1BlurAndNoise, filterPSF, fltNSR);
        imgTissue1Weiner3 = deconvwnr(imgTissue1BlurAndNoise, filterPSF, mxGaussianNoiseAutocorrelation, mxTissue1Autocorrelation);

        figure('Name', 'Wiener Filter', 'NumberTitle', 'off', 'MenuBar', 'none');
        colormap(gray);
        subplot(321);
        imagesc(imgTissue1);
        title('Original Image');
        subplot(322);
        imagesc(imgTissue1BlurAndNoise);
        title('Degraded Image');
        subplot(323);
        imagesc(imgTissue1Weiner1);
        title('Default Weiner Filter with NSR = 0');
        subplot(324);
        imagesc(imgTissue1Weiner2);
        title('Weiner Filter Using Noise to Signal Ratio');
        subplot(325);
        imagesc(imgTissue1Weiner3);
        title('Weiner Filter Using Autocorrelation with NSR');

%--------------------------------------------------------------------------
% Part 7 - Regularized Filtering
        imgTissue1Regular1 = deconvreg(imgTissue1BlurAndNoise, filterPSF);
        imgTissue1Regular2 = deconvreg(imgTissue1BlurAndNoise, filterPSF, fltGaussianNoisePower);  figure('Name','Regular Filter', 'NumberTitle', 'off', 'MenuBar', 'none');
        colormap(gray);
        subplot(221);
        imagesc(imgTissue1);
        title('Original Image');
        subplot(222);
        imagesc(imgTissue1BlurAndNoise);
        title('Degraded Image');
        subplot(223);
        imagesc (imgTissue1Regular1);
        title('Regular Filter With No Noise Power');
        subplot(224);
        imagesc(imgTissue1Regular2);
        title('Regular Filter Using Noise Power');

%--------------------------------------------------------------------------
% Part 8 - Lucy-Richardson Filtering
        imgTissue1LucyRichardson1 = deconvlucy(imgTissue1BlurAndNoise, filterPSF, 5);
        imgTissue1LucyRichardson2 = deconvlucy(imgTissue1BlurAndNoise, filterPSF, 3);

        figure('Name', 'Lucy-Richardson Filter', 'NumberTitle', 'off', 'MenuBar', 'none');
        colormap(gray);
        subplot(221);
        imagesc(imgTissue1);
        title('Original Image');
        subplot(222);
        imagesc(imgTissue1BlurAndNoise);
        title('Degraded Image');
        subplot(223);
        imagesc(imgTissue1LucyRichardson1);
```

```
        title('Lucy-Richardson Filter,
iterations=5');
        subplot(224);
        imagesc(imgTissue1LucyRichardson2);
        title('Lucy-Richardson Filter,
iterations=3');

%------------------------------------------------------------
--------------------
        disp('CPU Time:');
        disp((cputime - fltInitialCpuTime));

        disp('Done.');
%------------------------------------------------------------
--------------------
```

7. Simulation Results

Exploring the results of the restoration algorithms explained in section 5 as implemented in MATLAB.

Step 1

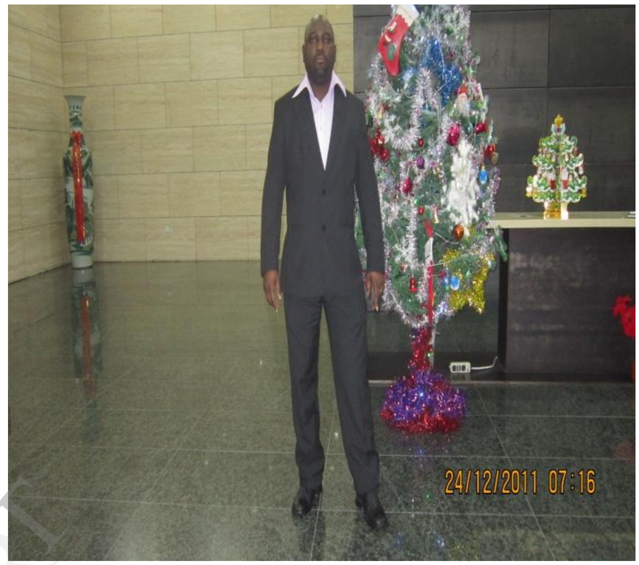
The original image, Gota.jpg is an RGB image and was converted to a grayscale image using the RGB2IND and IND2GRAY functions.

Fig.1: Original image (Gota.jpg)

Fig.2: Converted grayscale image (Gota_gray.jpg)

Step 2

A Gaussian Noise Matrix was generated using the RANDN function. The output of the noise matrix was intensity scaled and shown below in Figure 3. The intensities of the image matrix were scaled such that the highest intensity was 0.1 and the lowest: 0.0.



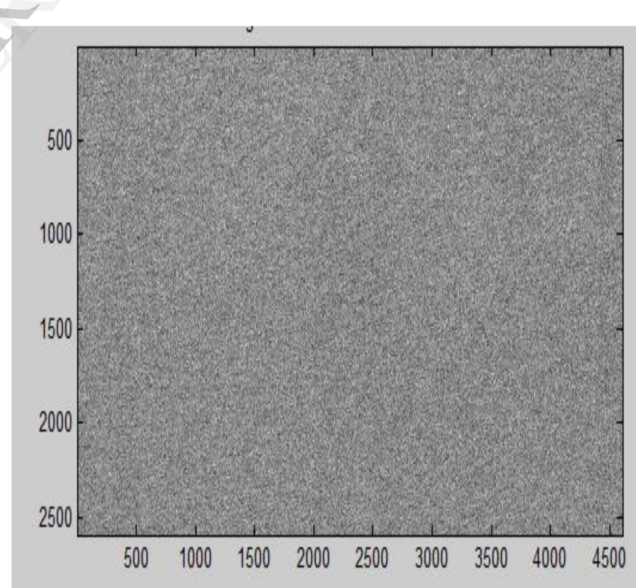


Fig.3: The Gaussian Noise Matrix scaled by intensity to 0 and 1 respectively

Step 3

A PSF was created using the FSPECIAL function using the motionparameter. The image in Figure 2 was then motion blurred using this PSF and the

IMFILTER function. Then, Gaussian noise was added with the IMADD function and the Gaussian Noise Matrix. The output of this operation on Figure 2 is shown below:

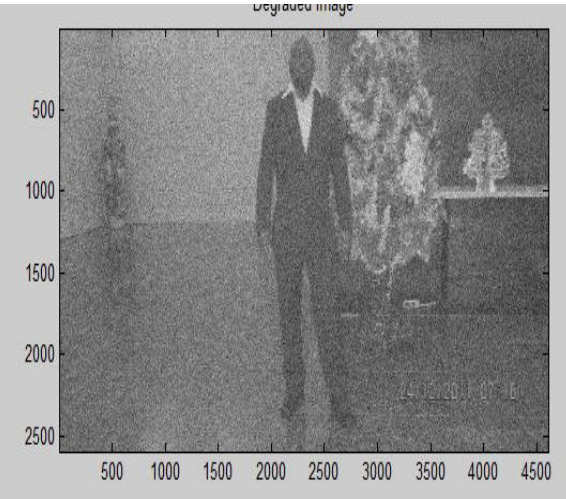


Fig.4: Degraded image of Figure 2 using motion blur and Gaussian noise

Step 4

The Noise to Signal Power Ratio was computed using the following equations:

Spectrum = abs (fft2 (image)). ^2;

Power= sum (Spectrum (:)) /prod (size (Spectrum)) ;

The equations were calculated for the image of Figure 2 and the image of Figure 3. The powers were then divided to get the Noise to Signal Power Ratio. The value of which is obtained as;

Noise to Signal Power Ratio: 0.0319

Step 5

The autocorrelation matrices were computed using the IFFT2 command. This returned the real part of the inverse fast Fourier transforms of the spectrums of figure 2 and figure 3. These matrices are used in the inverse filtering of Figure4 as would be seen later.

Step 6

The DECONVWNR function was used to perform Wiener filtering on Figure 3. There are many different parameters that can be used with this type of inverse filtering. Figure 5 shows these parameters being applied. Notice that the more information we give the filtering function, the better the results.

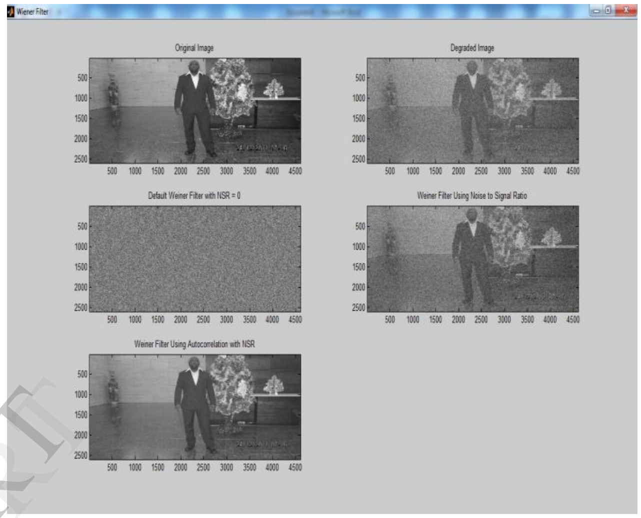


Fig.5: Results of Wiener filtering using different parameters.

Step 7

The DECONVREG function was used to perform Regulated filtering on Figure 3. There are many different parameters that can be used with this type of inverse filtering. Figure 6 shows these parameters being applied. Notice that the more information we give the filtering function, the better the results. With no noise power information, the image is heavily degraded. This is because the filter looks at the extra noise in the image as part of some degradation function blurring possibly and not as additive noise.

ISSN: 2278-0181
Vol. 2 Issue 12, December - 2013

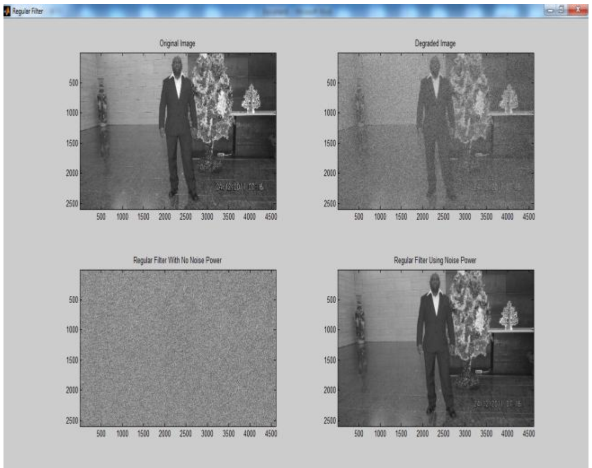


Fig.6: Results of Regulated filtering using different parameters.

Step 8

The DECONVLUCY function was used to perform Lucy-Richardson filtering on Figure 3. There are many different parameters that can be used with this type of inverse filtering. Figure 7 shows these parameters being applied. Notice that as the number of iteration we performed on the image increased, the better quality output image was realized. This filter is particularly interesting as it could remove much of the blur and still leave the noise in the image. This was not found with the two previous filters.

Fig.7: Results of Lucy-Richardson filtering using different parameters.

Same steps above where applied using motion blur but with Salt and Pepper and Speckle noise models respectively. The value of the Pepper and salt noise models respectively. The value of 0.05 for all the filters. Similarly, the variance v, for the speckle noise model was set at its default value of 0.04 for all filters. For the motion blur model, the linear motion of the camera LEN, with the angle of inclination THETA, were set at values of 21pixels and 11degrees respectively.

The following results were obtained:

A.Wiener filtering of salt and pepper noise

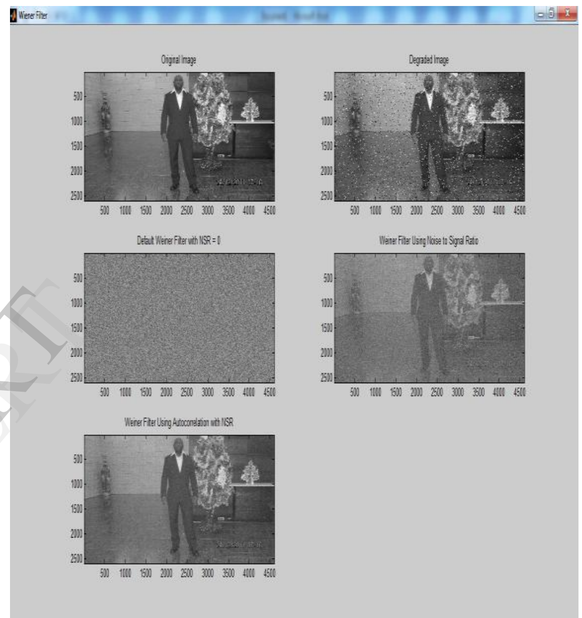
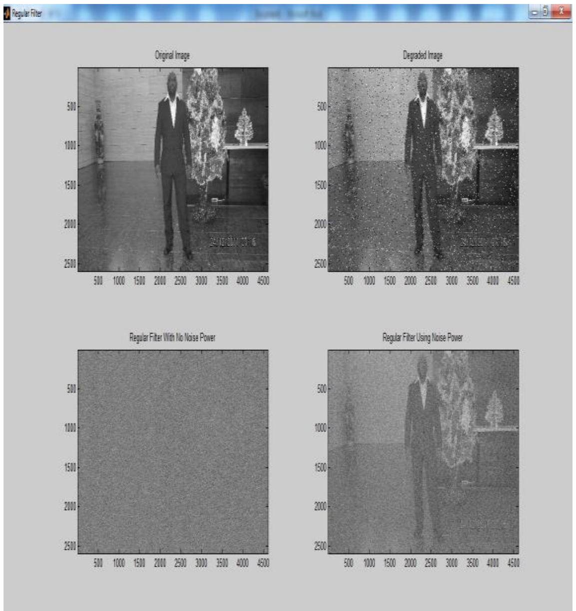


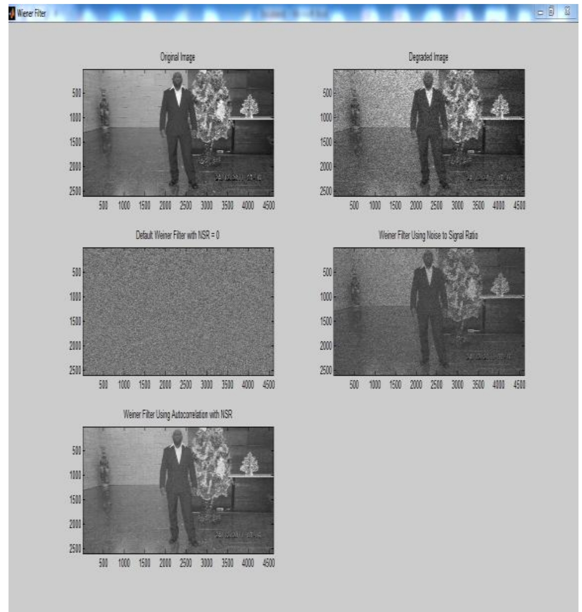
Fig.8: Results of Wiener filtering of Salt and Pepper noise using different parameters.

B.Regular filtering of Salt and Pepper noise

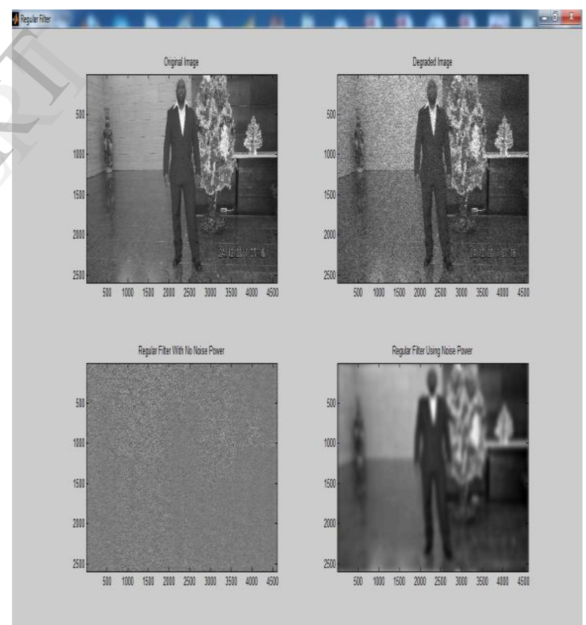


C.Lucy-Richardson filtering of Salt and Pepper noise

D.Wiener filtering of speckle noise



E.Regular filtering of Speckle noise



F.Lucy-Richardson filtering of speckle noise

7. Conclusion

In conclusion, often times the colour information if an image is irrelevant to analysis and when such is the case, a colour image is often converted to grayscale to speed up computation. The point-spread function (PSF) is unknown in real life and is often assumed based on camera parameters or other system characteristics. It can be computed by trial and error. However, the PSF is usually a model as it is very complex to determine an exact PSF.Although this paper tried exploring the effects of other noise models, noise in an image is commonly modelled as Gaussian. The noise to image power ratio is good measure of how much noise is in an image. This information, along with the noise power, can be used in the inverse filtering process to realize a much less degraded image. As expected the more information you add to the inverse filtering process, the better the output. Inverse filtering is achieved by the use of autocorrelation matrices that are obtained from noise and blur information about the image. This information expressed as a correlation is convenient way to implement it into an inverse filter. It can be observed from the results obtained that with no noise information, the Wiener and Regularized filters performance in realizing the degraded image was poor. However, the Lucy-Richardson filter had a good performance, despite having no information about the noise in the image. With noise information, the Wiener and Regularized filters did a great job at restoring the image. However, the Wiener filter is much better at the blur than the Regularized filter. Despite having no noise information, the Lucy-Richardson filter performs rather well at removing the degradation from the PSF (blur in the case) but not the noise. Therefore, having a good PSF, the Wiener and Regularized filters will perform better where the noise information is available whereas, the Lucy-Richardson filter performs better in blurs elimination and not particularly the noise.

8. References

1] www.owlnet.rice.edu/elec539/projects99

[2] www.ee.un/v.edu

[3] www.gerltd.com

[4] www.mathworks.com/help/toolbox/images/bqqhlbw.html

[5] A.S. Awad, *A Comparison between previously known and two novel Image Restoration Algorithms*, MS Thesis, Department of Electrical & Electronics Engineering, East Mediterranean University. June 2001.

[6]R. L. Lagendijk and J. Biemond, *Basic Methods for Image Restoration and*

Identification, Information and Communication Theory Group, Faculty of Information Technology and Systems Delft University of Technology, The Netherlands (February, 1999).

[7] R.C. Gonzalez and R.E. Woods, *Digital Image Processing (Third Edition)*, Publishing House of Electronics Industry, Beijing (2010).

[8] M. Jiang and G. Wang, *Development of Blind Image*

[9] M.R. Banham and A.K. Katsaggelos, Digital Image Restoration, IEEE Signal Processing Magazine 14(2) (March 1997), 24–41.

[10] M.I. Sezan and A.M. Tekalp, Survey of Recent Developments in Digital Image Restoration, Optical Engineering, vol. 29, No. 5, pp. 393-404, (1990).

[11] Pitas I., Digital Image Processing Algorithms, Prentice Hall, New York, (1993).

[12] Yung N.H.C., Lai A.H.C., Poon K.M., Modified CPI Filter Algorithm for Removing Salt-and-Pepper Noise in Digital Images, Proc. SPIE in Visual Communication and Image Processing'96, vol. 2727, (1996).

[13]X.Tai,O.Christiansen,P. Lin and I.Skjaelaaen. ARemark on theMBOSchemeandSomePiecewise ConstantLevelSetMethods,submitted.

[14]R. Glowinski, P. Lin andX.-B. Pan.Anoperator-splitting methodfor aliquid crystalmodel.CompPhysComm152,2003,pp.242-252.

[15]T. Lu,P. NeittaanmakiandX-CTai. Aparallelsplitting upmethodand itsapplication tonavier-stoke equations. AppliedMathematicsLetters, 4,1991,pp.25-29.

[16]T.Lu,P.Neittaanmaki, andX-CTai.Aparallelsplittingupmethodforpartial differentialequations anditsapplication to navier-stokes equations. RAIRO Math. Model.AndNumer. Anal.,26,1992,pp.673-708.

[17]J.Weickert, B.H.Romeny, andM.A.Viergever.Efficientandreliableschemesfor non- linear diffusionfiltering.IEEE Trans. ImageProcess,7,1998,pp.398-409,1998.